

Campo de Texto y Botón

Transcripción

[00:00] Hola amigos. ¿Cómo están? Vamos a continuar con nuestro curso de lógica de programación. Hasta este momento ya hemos visto bastantes cosas. Hemos aprendido a usar variables, hemos aprendido a diferenciar lo que es código HTML de código JavaScript en un programa, aprendimos a hacer algunas operaciones, operaciones matemáticas también y trabajar con cadenas de caracteres, string, en nuestro código.

[00:32] Aprendimos a hacer condicionales, if, else, y también a usar las iteraciones como for, while, inclusive iteraciones anidadas que es un for dentro de otro for, un while dentro de un while, en esa línea. Lo que vamos a aprender hoy día es a combinar, a convivir con código HTML y código JavaScript.

[00:59] Para ello, vamos a crear un nuevo programa parecido al de juego de adivinación, solo que lo vamos a llamar juego secreto. Vamos a empezar desde cero y aquí lo voy a, voy a dar un salvar como Juego Secreto.HTML. Ya les explico en qué consiste. Vamos a reutilizar de aquí solo la primer línea, recordando que esa primer línea es para que al momento de abrir en cualquier navegador nos acepten los caracteres especiales que son propios del lenguaje español.

[01:44] Vamos a reutilizar solo esto, recordando que, en la parte de arriba, en la primer parte siempre escribíamos el código HTML y después definíamos nuestro código de JavaScript con la siguiente etiqueta: script y escape. Solo para que nos situemos aquí.

[02:15] ¿Qué vamos a hacer? Existe un elemento en HTML que se llama input. Voy a salvar esto para mostrarles en qué consiste ese input y ya voy a ir explicándoles a ver cuál va a ser el objetivo de nuestro programa. Entonces, voy a abrir una nueva pestaña aquí en nuestro navegador, "Ctrl + O" para abrir Juego Secreto.

[02:44] Ese input lo que me da es un elemento en HTML donde el usuario puede ingresar cualquier valor. Recordando que en JavaScript habíamos visto el prompt, que aparecía un pop-up, un mensaje, una alerta y le decía al usuario con un campo para ingresar. Este de aquí es un poco más interesante porque no aparece ningún pop-up y aparece directamente en nuestra pantalla.

[03:17] Y tenemos otro elemento también de HTML que es el button, que lo que hace es crear un botón, solo que para este necesitamos abrir y cerrar, y aquí le vamos a poner: "verificar si acertó con el secreto". Entonces, lo que nuestro programa va a hacer, nuestro programa va a ser bien básico, vamos a ingresar un valor aquí en este input y un botón donde el usuario va a apretar para verificar si acertó con el secreto.

[03:54] Todo el secreto y el código lo vamos a definir aquí en JavaScript dentro de estas etiquetas. Vamos a guardar, save, y vamos aquí a nuestro juego. Actualizamos solo para ver cómo quedó. Entonces, tenemos aquí nuestro código y nuestro botón, que de momento no está haciendo nada.

[04:16] Volvemos aquí, y para iniciar nuestro programa lo primero que vamos a hacer es definir una variable, ya dentro de JavaScript, una variable que se llame secreto y le vamos a dar un número 5. Puede ser cualquier número o cualquier valor de cadena de texto.

[04:38] Vamos a definir una variable input, que va a ser nuestro comunicador entre el código JavaScript que está aquí y el código HTML que lo tenemos encima, recordando que le estamos pidiendo los valores y ejecuciones aquí en

el código HTML y tenemos que hacer que ese código HTML se converse con nuestro código JavaScript aquí. ¿Cómo hacemos eso?

[05:05] Así como teníamos el `document.write` donde le mandábamos en este caso a HTML: "escribime esto en la pantalla", tenemos una función que se llama `querySelector`. Aquí recordando, la `S` puntualizando siempre tiene que ser con mayúscula. `querySelector`, y le doy como parámetro el nombre de la etiqueta que quiero rescatar. En este caso, es `input`, `input`, que es esta etiqueta `input` aquí.

[05:47] Entonces, lo que va a hacer esto es traerme todo este objeto de `input` aquí. Entonces, lo que me queda es hacer un `if`, vamos aquí a estructurarlo y un `else`. ¿Qué vamos a ponerle aquí en el `if`? Si `secreto` igual a `input`, vamos a decirle que acertó. Si `secreto` diferente de `input`, vendría a ser el `else`, le vamos a decir que erró.

[06:21] Primeramente, para hacer nuestra condición, no puedo comparar `input` directamente porque en este caso `input` es un objeto que me lo está capturando. Me está capturando toda la etiqueta. Tengo que pasarle `input.value`. Solo que también aquí tengo que cerciorarme, convertir ese número a entero.

[06:44] Ya lo habíamos visto eso, porque voy a comparar un valor que va a ingresar el usuario, nos vamos a cerciorar que sea número, porque nuestro `secreto` está definido como un número, y ese `input` del usuario lo vamos a comparar con nuestra variable `secreto`.

[07:09] Entonces, estamos preguntando aquí: si lo que ingresa el usuario es igual a `secreto`, entonces que aparezca una alerta y que le diga a nuestro usuario: "Usted acertó". Si el usuario no acertó, que le diga: "Usted erró". Punto y coma aquí, punto y coma aquí. Voy a salvarlo, aquí lo vamos a guardar, `save` y vamos a nuestro programa. Y ya nos dice: "Usted erró". ¿Por qué?

[07:57] No me deja ni colocar nada. Ya de entrada me dice que yo erré. Vamos a analizar nuestro código. Entonces estoy definiendo, estoy capturando aquí el input y ya automáticamente lo está comparando. O sea, me está comparando vacío, porque yo no ingresé nada cuando le doy a actualizar, ya me está comparando vacío con el valor secreto que es 5. Es por eso que me dice que yo erré.

[08:22] Lo que necesitamos es crear algún trigger, algún gatillo, que va a ser este botón, donde me va a comparar el valor que está ingresado aquí con mi valor secreto. Mirando el código secuencial y la ejecución secuencial como la ejecuta, no hay ningún break aquí, ningún lugar donde yo le diga: "comparame lo que ingresé en este input con la variable secreto."

[08:55] Entonces, automáticamente ejecuta todo, y por eso es que me dice que yo erré, porque me está comparando vacío con secreto. Entonces, para ello, vamos a definir una función, function, que la vamos a llamar verificar. No va a tener ningún parámetro, y dentro de nuestra función vamos a colocar todo esto.

[09:33] Bueno, vamos a dar aquí un save, guardar, y vamos a verificar allá nuestro código. Bueno, ya no me aparece el error, si aprieto aquí algo, el botón todavía no está funcionando porque no le he dicho que me haga nada con este botón, no le he dado ninguna instrucción. Solo que para entender un poco nuestro código voy a abrir aquí las herramientas de desarrollador.

[10:00] Y fíjense, he colocado cuatro. Si llamo aquí a la función ya verificar, él me dice: "Usted erró" porque no estoy llamando a la función en ningún momento. Lo único que hice fue crear la función que está ahí esperando que alguien la llame. Erré porque el número era 5 y yo he colocado 4. Voy a colocar 5. Verificar. Y ya me dice: "Usted acertó".

[10:28] Obviamente que el usuario va a tener que entrar cada vez aquí a la herramienta de desarrollador a hacer eso, no tiene ningún sentido. Entonces,

lo que vamos a hacer es, ni bien termina la función, vamos a crear una variable `button`, de botón, y vamos a ocupar esto mismo, o sea el `querySelector`, para extraer este elemento, este objeto, y ver las acciones que estamos haciendo con este objeto.

[11:08] Entonces, variable `button`, y aquí el nombre de mi etiqueta allá arriba que es `button`. Solo que hasta aquí lo único que he hecho es extraer, capturar el objeto `button` que lo definí aquí en HTML. Entonces lo que voy a hacer para `button`, al momento, click on, al momento que el usuario haga clic en ese botón, que me llame a verificar. Abro, cierro y pongo punto y coma.

[11:56] Voy a guardarlo, voy a volver aquí y vamos a volver aquí.

Automáticamente me dice: "Usted erró". Aquí también hay un pequeño detalle que necesitamos estudiarlo. Fíjense que le estoy dando verificar ya de entrada, o sea, carga todo este código en el compilador y ya yo le estoy diciendo que vaya y me verifique.

[12:25] No le estoy diciendo: "hay una pequeña, hay una gran diferencia, la verdad", aquí les voy a mostrar. Cuando yo llamo a verificar, me ejecuta todo ese código, pero cuando yo llamo a verificar sin la abertura y el cierre de paréntesis, lo que hace es traerme todo el código.

[12:53] Entonces, lo que vamos a hacer aquí es quitarle los dos paréntesis para que no lo ejecute de entrada sino lo ejecute al momento que yo haga clic en el botón. Vamos a guardarlo y vamos a volver aquí. Ahora sí, voy a colocar un 3. Aquí debo tener algún pequeño error en mi código, vamos a verificarlo, aquí me está comparando. Hasta aquí todo está okay.

[13:39] La función verificar. Voy a guardarlo y vamos a ver aquí. Vamos a hacer una prueba con 4 y aquí vamos a llamar a verificar. "Usted erró". Vamos a colocar 5. "Usted acertó". Entonces, el problema está con nuestro botón. En algún lugar nuestro botón no lo estamos capturando. `Button`, click on. Aquí en

verdad es on click, ese era el problema. Aquí lo vamos a guardar, vamos a volver aquí.

[14:54] Actualizamos y ahora sí. 4. "usted erró". 5, "usted acertó". Entonces 1, "usted erró". 6, "usted erró". 5 nuevamente, "usted acertó". Entonces, nuestro programa está funcionando perfectamente. Aquí el error era que había colocado click on, pero es on click, como diciendo al momento de hacer click, y que me llame a la función verificar.

[15:33] Con esto, amigos míos, hemos aprendido a comunicar, a hacer un puente entre el código JavaScript y el código de HTML, haciendo uso de esta función que es el querySelector. Por favor hagan los ejercicios, acompañen todo el código que estamos haciendo y nos vemos en la próxima. Muchas